13701 Postel

NIC# 13701

RFC# 437

RECEIVED
AUG 9 1973

COMPUTER SYSTEMS
LABORATORY--UCSB
Ed Faeh
30 June 1973

DATA RECONFIGURATION SERVICE AT UCSB

Postel

The purpose of this RFC is to announce the availability of the Data Reconfiguration Service (DRS) at UCSB, and to describe the use of the DRS Time Sharing System. The DRS is an experiment in a flexible means for reformatting Network data streams. The DRS provides a means for coupling processes with different input/output interfaces, and carrying out user specified transformations on the data passing between them. Samples of representative uses of the DRS include field insertion, field deletion, variable length string processing, string length computation, field transposition, character packing and unpacking, and character set translations.

To use the DRS, a user first defines a "form", or a description of the reformatting to be performed on data passing between two sockets (a form is associated with each unidirectional message path). DRS may then be directed to establish Network connections with the two processes involved and to monitor the dialogue between them. DRS receives an input stream from one process, reformats the input according to the rules specified by the form, and emits the reformatted data as an output stream to the second process. The two processes communicate as if they were directly connected to each other.

Three major components comprise the DRS: a compiler which reduces DRS source programs (forms) to a simpler, machine independent instruction sequence (object program), an interpreter which executes the object program created by the compiler, and an executive program, the DRS Time Sharing System, which interfaces the Network user to the DRS. Detailed descriptions of the DRS source language and compiler are available in the following documents:

> "The Data Reconfiguration Service--An Experiment in Adaptable, Process/Process Communications", The Rand Corp., R-860-ARPA, November 1971.
>
> "Data Reconfiguration Service Compiler: Communications Among Heterogeneous Computer Centers Using Remote Resource Sharing", The Rand Corp., R-887-ARPA, April 1972.

The DRS Time Sharing System (DRS/TSS) and its server Telnet are currently available and addressable through socket 1281 decimal. DRS/TSS interfaces the user to the DRS. In addition to a subset of 'TENEX-like' executive commands, commands are available for creating and storing forms by name, and for invoking the DRS compiler or interpreter. Since both the compiler and the interpreter run asynchronously to the DRS/TSS, the user is notified when execution of his requests for compilations or interpreting are initiated and terminated. In addition, a diagnostic message is supplied by the interpreter whenever it terminates execution of a form.

RECEIVED AT NIC 8-3-73

When a user connects to DRS/TSS he is allocated an internal TTY number by which his terminal is identified. When the user issues a command to invoke the compiler or interpreter, his TTY number and a user specified job name are used to uniquely identify the request. The jobname is thereafter used to refer to the request. A Network host is specified by host name, or by host number preceded by #. A socket number is specified as an integer, optionally preceded by a plus or minus sign, and an H or X (hexadecimal), O (octal), or D (decimal) to indicate the radix used. Decimal is the default radix and positive the default sign. A negative socket number is maintained in two's complement form.

The following control characters are recognized by DRS. Control A (SOH) or rubout (DEL) delete the last character, control V (SYN) deletes the last word, control W (ETB) deletes the last phrase, and control X (CAN) deletes the entire line. Control D (EOT) displays the current word, control E (ENQ) displays the current phrase, and control F (ACK) displays the entire line. Control Y (EM) displays the name of the current phrase. Command accept is carriage return-line feed, or period. A space will force recognition of the current word, and '?' will list the input(s) currently possible.

The following sections describe the DRS/TSS command syntax. <CA> denotes command accept, and brackets [ ] denote optional information. All literal information is enclosed in quotes ' '. Choose one alternative from items in parenthesized lists separated by | .

DRS/TSS COMMAND SYNTAX

Commands:

'ATTACH'	<tty no.> <password> <CA>
'BREAK'	(<tty list>|<CA>) <CA>
'COMPILE'	<job name> <formid> <CA>
'CONVERT'	<integer> ['TO'] <base> <CA>
'DEFINE'	<formid> <CA>
'DELETE'	[<file type>] <formid> <CA> <CA>
'DETACH'	<CA> <CA>
'DIRECTORY'	<formid> <CA>
'DISCONNECT'	<CA> <CA>
'FULLDUPLEX'	<CA>
'HALFDUPLEX'	<CA>
'HELP'	<CA>
'LINE_AT_A_TIME'	<CA>
'LINK'	('ENABLE'|'DISABLE'|<tty list>|<CA>) <CA>
'LIST'	[<file type>] <formid> <CA>
'LOGIN'	<userid> <password> <account number> <CA>
'LOGOUT'	<CA>

```
'MONITOR'           (<tty list>|<CA>) <CA>
'QUIT'              (<job name>|<CA>) <CA>
'RADIX'             <base> <CA>
'RENAME'            <formid> ['TO'] <formid> <CA>
'RESPONSE'          ('ALL'|'HANDHOLD'|'PROMPT'|'NORMAL'|'UNUSUAL'|
                     'ERRORS'|'NONE') <CA>
'RUN'               <job name> ('SIMPLEX' <run parameters>|
                     'DUPLEX' '(' <run parameters> <run parameters>
                     ')' ) <CA>
'SELECT'            (<tty list>|<CA>) <CA>
';'                 <text> <CA>
```

Substitutions:

```
<formid> = ['<' <userid> ',' <password> '>'] <form name>
<base> = ('DECIMAL'|'HEXADECIMAL'|'OCTAL')
<file type>  =('DIAGNOSTICS'|'OBJECT_CODE'|'SOURCE')
<run parameters>  = ('SICP' <socket number>|'UICP'
                    [<hostid>] <socket number>|'RFC'[<hostid>]
                    <socket number> [<hostid>] <socket number>)
                    ['BYTE SIZE' <integer>]  ['CONNECT_TIME'
                    <integer>]  ['RUN_TIME' <integer>]  <formid>
```

Definitions:

<acct no>	Account number is a character string whose maximum length is 4.
<form name>	Form name is a character string whose maximum length is 6, and whose first character is alphabetic.
<hostid>	Host number preceded by '#' or host name may be used to specify a Network host.
<integer>	Integers may optionally be preceded by a plus or minus sign, and by an H or X (hexadecimal), O (octal), or D (decimal) to specify the radix used. Decimal is the default radix and positive the default sign.
<job name>	Job name is a character string whose maximum length is 6, and whose first character is alphabetic.
<password>	The currently acceptable password is 'ARPA'.
<socket number>	Socket numbers are specified as <integer> (see above).

<tty list> TTY List is one or more TTY numbers, separated by commas.

<tty no.> TTY number is an integer representing a DRS terminal identification number.

<userid> User ID is a character string whose maximum length is 8, and whose first character is alphabetic.

Command Descriptions:

'ATTACH' <tty no.> <password> <CA>

Connects the user's console to the DRS/TSS terminal with identification number <tty no.>. Terminal <tty no.> must have been previously detached, and must not have been deleted. (Currently, detached terminals are deleted after 1 hour.)

'BREAK' (<tty list>|<CA>) <CA>

Causes links to the terminals specified in <tty list> to be broken. If <tty list> is not specified, all established links are broken. (See 'LINK', 'MONITOR', and 'SELECT' for a description of how links are established.)

'COMPILE' <job name> <formid> <CA>

Causes form <formid> to be compiled. Verification that the source for form <formid> exists, and that object code does not already exist is made before invoking the DRS Compiler, which executes asynchronously to the user's terminal. The DRS Compiler generates a diagnostic file for the form and, if no errors are encountered, object files. <job name> is used to identify the request and may not be used to identify any other job so long as the current request is outstanding.

Because of main storage requirements, only one compilation is allowed in execution at any time. Subsequent requests are queued for later execution unless the request queue is full, in which case the request is refused. The user is notified when his compilation is initiated, and when it terminates.

'CONVERT' <integer> ['TO'] <base> <CA>

Allows the user to convert octal, decimal, or hexadecimal integers from one radix to another. <base> specifies the desired radix of the result and may be 'OCTAL', 'DECIMAL', or 'HEXADECIMAL'.

'DEFINE' <formid> <CA>

Allows the user to define a new source form, or append to an existing source form. The new line indicator '$$' replaces the normal '$' to indicate that input will be used to define the source form. Control C (ETX) signals the end of form definition and elicits the normal '$' new line indicator.

Unfortunately, the current facilities for editing forms are extremely limited. Only the line currently being entered may be edited. Once a line has been terminated, with CR-LF, and the new line indicator has appeared, the line may not be edited. The following control characters may be used to edit a form while it is being defined. Control A (SOH) or Rubout (DEL) delete the last character, control X (CAN) deletes the entire line, and control F (ACK) displays the entire line. If modifications need to be made to lines other than the current line, the entire form must be deleted and reentered.

'DELETE' [<file type>] <formid> <CA>

Deletes the specified component file(s) of form <formid>. <file type> specifies the component file to be deleted, either 'SOURCE', 'DIAGNOSTIC', or 'OBJECT_CODE'. If <file type> is not specified, all components of the form are deleted.

'DETACH' <CA> <CA>

Breaks the TELNET connection between DRS and the user, but does not. delete the user's DRS terminal for some fixed period of time, currently 1 hour. Any jobs the terminal has active are unaffected when the terminal is detached. The user may subsequently reconnect to his DRS terminal by using the 'ATTACH' command.

'DIRECTORY' <formid> <CA>

Lists which component(s) of form <formid> exist. A complete form consists of 'SOURCE', 'DIAGNOSTICS', and 'OBJECT_CODE' files.

'DISCONNECT' <CA> <CA>

Breaks the TELNET connection between DRS and the user and deletes the user's DRS terminal. Any jobs the terminal has active are purged.

'FULLDUPLEX' <CA>

Directs DRS to echo user input; nullifies a previous 'LINE_AT_A_TIME' command.

'HALFDUPLEX' <CA>

Directs DRS not to echo user input; nullifies a previous 'LINE_AT_A_TIME' command.

'HELP' <CA>

Lists the DRS control characters and their meanings.

'LINE_AT_A_TIME' <CA>

Tells DRS that the user is operating in line-at-a-time mode, and nullifies a previous 'FULLDUPLEX' or 'HALFDUPLEX' command. DRS will not send any output to the user while he is inputting commands.

'LINK' ('ENABLE'|'DISABLE'|<tty list>|<CA>) <CA>

The link command may be used to enable or disable links to other terminals, or to establish links between terminals.

'LINK' 'ENABLE' <CA> directs DRS to allow subsequent requests by other terminals to establish links with the user's terminal.

'LINK' 'DISABLE' <CA> directs DRS to refuse subsequent requests by other terminals to establish links with the user's terminal.

'LINK' (<tty list>|<CA>) <CA> requests DRS to establish bidirectional links between the user's terminal and the terminals specified in <tty list>. If <tty list> is omitted, DRS attempts to link the user's terminal to all other active terminals. While the user's terminal is linked to another terminal, all output to the user's terminal will also be routed to the linked terminal, and all output to the linked terminal will also be routed to the user's terminal. Unidirectional links may be established via the 'MONITOR' and 'SELECT' commands.

'LIST' [<file type>] <formid> <CA>

Lists the contents of the <file type> file of form <formid>. <file type> may be 'SOURCE', 'DIAGNOSTICS' or 'OBJECT_CODE', and defaults to 'SOURCE'.

'LOGIN' <userid> <password> <acct no.> <CA>

Logs the user in, enables job-oriented commands, and specifies the account to which the use of DRS is to be billed. Currently billing for DRS is not implemented, and <userid> and <acct no.> may be arbitrarily chosen by the user. When billing for DRS is implemented, user ID's and account numbers will be assigned and checked for validity at login. The only currently acceptable <password> is 'ARPA'.

'LOGOUT' <CA>

Logs the user out and disables job-oriented commands and billing of subsequent activity to the previously specified accounting parameters. As with 'DISCONNECT', any jobs the terminal has active are purged.

'MONITOR' (<tty list>|<CA>) <CA>

Requests DRS to establish unidirectional links between the user's terminal and the terminals specified in <tty list>. If <tty list> is omitted, DRS attempts to link the user's terminal to all other active terminals. While the user's terminal is monitoring another terminal, all output to that terminal will also be routed to the user's terminal.

'QUIT' (<job name>|<CA>) <CA>

Requests DRS to purge the job (request for compilation or interpreting) identified by <job name>. If <job name>is omitted, all active jobs are purged.

'RADIX' <base> <CA>

Allows the user to specify the radix to be used for subsequent integer display. <base> may be 'OCTAL', 'DECIMAL', or 'HEXADECIMAL'. The default display radix is decimal.

'RENAME' <formid> ['TO'] <formid> <CA>

Allows the user to rename a form. The names of all components of the form specified by the first <formid> are changed to correspond to the second <formid>. If the form specified by the second <formid> already exists, no change of name occurs, and the user is notified.

'RESPONSE' ('ALL'|'HANDHOLD'|'PROMPT'|'NORMAL'|'UNUSUAL'|'ERRORS'|
'NONE') <CA>

Allows the user to specify the level of messages the system should provide. The default response level is 'ALL'.

'RUN' <job name> ('SIMPLEX' <run parameters>|'DUPLEX' '(' <run parameters>
<run parameters> ')') <CA>

Causes DRS to establish the requested Network connections and direct the DRS interpreter to apply the specified form(s) to data passing over the connections. The user first specifies the job name by which his request will be identified, and second whether simplex or duplex Network connections are to be established. 'SIMPLEX' implies that a single unidirectional message path is to be employed, and requires one set of <run parameters>. 'DUPLEX' implies that two message paths will be employed and requires two sets of <run parameters>. <run parameters> specify the data necessary to establish a Network connection, time limits for the DRS interpreter, and the form which is to be applied to the Network data stream.

```
<run parameters> = ('SICP' <socket number>|'UICP' [<hostid>]
                   <socket number>|'RFC'  [<hostid>] <socket
                   number> [<hostid>] <socket number>)
                   ['BYTE_SIZE' <integer>] ['CONNECT_TIME'
                   <integer>] ['RUN_TIME' <integer>] <formid>
```

Three methods are available for establishing Network connections: 'SICP', 'UICP', and 'RFC'. 'SICP' (Server Initial Connection Protocol) and 'UICP' (User ICP) direct DRS to use the standard Initial Connection Protocol. 'SICP' causes DRS to simulate a server process addressable through socket <socket number> at UCSB and wait for a user process to connect to it. 'UICP' directs DRS to simulate a user process and connect to the server addressable through Host <hostid>, socket <socket number>. 'RFC' directs DRS to connect directly to the two sockets specified. In the case of 'RFC', one of the two sockets specified must be a send socket, and the other must be a receive socket. <hostid> defaults to the Host from which the user logged into the DRS/TSS.

The remaining fields in <run parameters> are defined as follows:

'BYTE_SIZE <integer>	specifies the byte size to be used, and defaults to 8.
'CONNECT_TIME <integer>	specifies the amount of time, in seconds, that DRS should wait for the connection to be established. The default value is 30 seconds.
'RUN_TIME' <integer>	specifies the maximum amount of time, in seconds, that the DRS interpreter will be allowed to execute. The default value for 'RUN_TIME' is five minutes.
<formid>	specifies the form which is to be applied to the Network data stream.

For each instance of <run parameters>, one socket pair is allocated and one form is specified. In the case of 'SIMPLEX' runs, the Network data flows between these two sockets, with the form being applied to the data. In the case of 'DUPLEX' runs, the connections are somewhat more complex. For 'DUPLEX' runs, two sets of <run parameters> are specified, causing two socket pairs to be allocated, and two forms to be specified. Call these send socket 1, receive socket 1, and form 1 (established via the first set of <run parameters>) and send socket 2, receive socket 2, and form 2 (established via the second set of parameters). When interpreting begins, data will flow from send socket 1, through form 1, to receive socket 2,and from send socket 2, through form 2, to receive socket 1. (Note that without this socket switching, processes would be connected to themselves when 'SICP' or 'UICP' is specified.)

The user is notified when his requests for connections are initiated and when the interpreter begins applying a form to a connection. When execution of a form terminates, the user is supplied with a diagnostic message provided by the interpreter as well as the actual run time of the interpreter.

'SELECT' (<tty list>|<CA>) <CA>

Requests DRS to establish unidirectional links between the user's terminal and the terminals specified in <tty list>. If <tty list> is omitted, DRS attempts to link the user's terminal to all other active terminals. While the user's terminal has another terminal selected, all output to the user's terminal will also be routed to the selected terminal.

';' <text> <CA>

Allows the user to enter any comments he chooses. <text> may include the characters '?', and '.'. Note that <CA> must be CR-LF, rather than period.